



Desenvolvimento de Jogos Multiplayer



Os modelos de rede para jogos multijogador descrevem a arquitetura e o design usados para estabelecer a comunicação entre a fonte de informação e o destino – como entre um cliente e um servidor ou entre pares (ARRUDA, 2014).

De acordo com Unity Technologies (2023), podemos relacionar a definição do servidor e cliente a partir dos seguintes elementos:

Como alternativa, um player também pode receber uma função de servidor (ou host) em uma topologia ponto a ponto típica.

Já um modelo ponto a ponto (P2P) conecta todos os jogadores diretamente, sem a necessidade de um servidor ou host dedicado. Ao contrário dos modelos cliente-servidor, que exigem um servidor atribuído que detém a palavra final no estado do mundo, em um modelo P2P, o dispositivo de cada jogador é responsável por executar a lógica do jogo e negociar com os outros (MINOR, 2022). Uma rede P2P pode ser:

Os modelos de servidor definem qual “expressão” do modelo cliente-servidor é usada e ditam quem é o “servidor” na relação cliente-servidor. Os dois principais modelos de servidor são:

Há uma variedade de modelos de servidor que seu jogo multijogador pode escolher, e cada um é adequado para diferentes gêneros de jogos. Um servidor hospedado pelo cliente (também conhecido como servidor de escuta) é um modelo no qual a lógica do jogo é hospedada em uma das

próprias máquinas do jogador, em vez de utilizar um servidor dedicado separado. A máquina host atua tanto como servidor quanto como cliente.

Isso significa que um dos jogadores jogará e será o dono do mundo do jogo e outros jogadores se conectarão a ele. Esse modelo tem considerações de desempenho, segurança e confiabilidade a serem levadas em consideração.

Um modelo de servidor dedicado é onde a lógica do jogo é hospedada em uma máquina separada dos jogadores. Servidores dedicados simulam mundos de jogo sem suportar entrada ou saída direta, exceto o que é necessário para administração. Esses servidores atuam como a máquina de estado onde toda a telemetria de objetos em rede dentro da simulação é reconciliada (MINOR, 2022).

Servidores de jogos dedicados também podem compensar a latência da rede (atraso) usando recursos de código de rede, como interpolação e previsão. Os jogadores devem se conectar ao servidor com programas clientes separados para ver e interagir com o jogo; o servidor do jogo é a autoridade de onde tudo fica dentro da simulação e onde está a qualquer momento. Como a lógica é hospedada em um dispositivo separado dos jogadores/clientes, um modelo de servidor de jogo dedicado é mais seguro, pois os IPs dos jogadores que se conectam a esse servidor não seriam expostos a um oponente como seriam no modelo hospedado pelo cliente.

Tipo de plataformas de jogo multijogador

Os jogos multiplayer podem ser divididos ou categorizados por diferentes fatores. Existem duas plataformas populares nas quais muitos desenvolvedores trabalham: Android e iOS.

Os desenvolvedores de jogos Android desfrutam de grande liberdade para acessar inúmeras plataformas e ferramentas que tornam o fluxo de trabalho

organizado. Algumas das principais linguagens de programação usadas no desenvolvimento de jogos multiplayer para Android são Java, Kotlin, C C++. A melhor parte de usar esta plataforma é que a certificação e aprovação de aplicativos é fácil e simples para Android. Ela possui diretrizes flexíveis e um sistema relativamente transparente, o que permite que a maioria dos aplicativos seja publicada em até 2 dias após o envio.

O desenvolvimento de jogos multiplayer para iOS é amplamente conhecido por sua uniformidade e padronização. A principal programação usada pelos desenvolvedores nesse segmento é Swift e Objective C. Com esse tipo de desenvolvimento, você pode atingir um público relativamente maior em todo o mundo, que é de cerca de 1,65 milhão de usuários. No entanto, o custo dos jogos multijogador iOS será comparativamente maior do que o do Android. Além disso, o processo de aprovação e certificação de aplicativos é bastante complexo e difícil nesse tipo de plataforma de desenvolvimento de jogos (UNITY TECHNOLOGIES, 2023).

Por fim, podemos observar sobre o processo de desenvolvimento de jogo multijogador multiplataforma. Esse tipo de recurso é usado principalmente por desenvolvedores para projetar e desenvolver aplicativos de jogos multijogador de ponta a ponta de alto valor. Isso ajudará a atingir não apenas o público internacional, mas também o doméstico.

Atividade Extra

A partir do que foi apresentado neste módulo, principalmente no processo de desenvolvimento de jogos digitais multiplayer, recomenda-se a leitura do artigo científico “Software anti-cheat: uma maneira de prevenir trapaças em jogos multiplayer” (VICENTINE, A. L.; SILVA, G. C. M.; NASCIMENTO, J. P. D.

G.; NEVES, J. E. D. A.; 2022), disponível no Google Acadêmico e no diretório da Faculdade de Tecnologia de Araraquara.

Será possível observar sobre os princípios de segurança em jogos multiplayer, relatando sobre cheats e softwares de trapaças que concedem vantagens aos adversários. Esse é um dos maiores problemas que os usuários de jogos online enfrentam. Além disso, serão mencionados os softwares anti-cheat, cujo objetivo é detectar trapaças em jogos online com múltiplos jogadores.

Referência Bibliográfica

ARRUDA, E. P. **Fundamentos para o Desenvolvimento de Jogos Digitais**. Porto Alegre: Grupo A, 2014.

MINOR, J. **The Best Multiplayer Video Games for 2023**. 2022. Disponível em: < https://www.pcmag.com/picks/the-best-multiplayer-video-games >. **(Acesso em 19 de Abril de 2023)**

RABIN, S. **Introdução ao Desenvolvimento de Games**. São Paulo: Cengage Learning Brasil, 2012.

UNITY TECHNOLOGIES. **An Introduction To Multiplayer Network And Server Models**. 2023. Disponível em: https://unity.com/how-to/intro-to-network-server-models **(Acesso em 19 Abril de 2023).**

- **Servidor:** Uma entidade que detém a palavra final sobre qual é o estado atual do mundo.

- **Cliente:** É uma entidade que compartilha sua versão do estado do jogo/mundo com o servidor. Todos os dispositivos conectados possuem uma versão do mundo, e o principal problema resolvido com os modelos de rede multijogador é que uma solução de código de rede está tentando oferecer a mesma experiência a todos os jogadores conectados da maneira mais suave possível.

- **Modelo cliente-servidor:** É um modelo em que existe um "servidor" que envia informação aos "clientes" que consomem essa informação.

- **Modelo peer-to-peer:** É um modelo no qual todos os jogadores estão conectados diretamente, sem a necessidade de um servidor.

- A arquitetura cliente-servidor é um modelo em que existe um "servidor" que hospeda a lógica do jogo e envia informações para os "clientes" que consomem essas informações e agem de acordo. Os clientes também podem enviar dados para o servidor, como entradas e ações. A partir desse contexto, de acordo com RABIN (2012), podemos relacionar que:

- Qualquer provedor de informações ou serviços é referido como um servidor.

- Os solicitantes de serviços e receptores dessas informações são referidos como clientes.

- O servidor, neste cenário, é provavelmente um servidor de jogo dedicado, puramente focado em simular o estado do jogo e retransmitir informações.

- Uma conexão ad hoc: são dois dispositivos conectados para compartilhar arquivos.

- Uma infraestrutura permanente que liga alguns computadores em um pequeno escritório por fios de cobre.

- Uma rede em uma escala muito maior na qual protocolos e aplicativos especiais estabelecem relacionamentos diretos entre os usuários pela Internet.

- **Hospedado pelo cliente:** É um modelo em que um jogador no jogo atua como o host e seu dispositivo é a autoridade no estado do jogo.

- **Servidor de jogo dedicado:** É um modelo em que a simulação do jogo é executada em um dispositivo separado. Esse dispositivo é a autoridade no estado do jogo, mas não está jogando ou renderizando o jogo.

Atividade Prática

Título da Prática: Desenvolvimento de Jogos Multiplayer.

Objetivos: Analisar o processo de desenvolvimento de Jogos Multiplayer.

Materiais, Métodos e Ferramentas: Browser (navegador), ferramenta de edição de textos e a game engine Unity 3D.

Roteiro

1º. Passo) Leia atentamente o texto:

A Unity Multiplayer refere-se à "**Unity Multiplayer Networking**", que é o conjunto de ferramentas Unity para desenvolvimento de jogos multiplayer. Em 2021, a Unity deu um grande passo em direção à rede multijogador. Foi quando eles lançaram oficialmente o Netcode para GameObjects, oferecendo um recurso estendido focado em jogos cooperativos de ritmo moderado e pequena escala. Ao mesmo tempo, a Unity aprimorou o Unity Transport e adicionou suporte para criptografia DTLS. Além disso, naquela época, a equipe da Unity lançou um novo recurso Network Profiler e serviços multijogador – Unity Relay e Lobby.

No entanto, a partir de 2022 os desenvolvedores da Unity não precisaram mais contar com ferramentas de terceiros para oferecer suporte a cer tipos de jogos multijogador. Desde então, os desenvolvedores tiveram acesso ao Netcode for GameObjects (NGO) 1.0. Essa versão está alinhada com as novas soluções multiplayer da Unity, incluindo Game Server Hosting (Multiplay) e as soluções de matchmaker.

Essa atualização faz parte do objetivo da Unity de contribuir para o surgimento de jogos online com uma série de soluções prontas para uso. No entanto, vale ressaltar que a Unity ainda permite que os desenvolvedores misturem e combinem suas soluções com ferramentas de terceiros para encontrar a melhor pilha de tecnologia para seus projetos.

2º. Passo) Acesse o site oficial da documentação da Unity UMultiplayer Networking para obter maiores informações sobre a utilização do Netcode:

O Netcode for GameObjects (Netcode) é uma biblioteca de rede de alto nível criada pela Unity para que os programadores possam abstrair a lógica de rede. Esse recurso permite que os usuários enviem GameObjects e dados mundiais através de uma sessão de rede, possibilitando a interação de muitos jogadores simultaneamente.

Com o Netcode, é possível se concentrar em criar o jogo em vez de protocolos de baixo nível e estruturas de rede.

3º. Passo) Observar o processo de instalação e vinculação da biblioteca Netcode na Unity:

O Netcode suporta a versão 2020.3 e superior. O Netcode não suporta a plataforma WebGL, pois não permite acesso a Sockets IP.

Para instalar o Netcode na Unity, devemos realizar as seguintes ações:

1. Na barra de menus, selecione Windows → Gerenciador de pacotes.

2. Clique no sinal de mais (+) e selecione adicionar pacote por nome.



3. Copie e cole o seguinte: com.unity.netcode.gameobjects

4. Clique em Adicionar.

A primeira coisa que precisamos adicionar à nossa cena é um gerenciador de rede. O NetworkManager é um componente Netcode necessário para a manipulação dos GameObjects que contém todas as configurações relacionadas ao netcode do projeto. Para adicionar isso à nossa cena (dentro do projeto da Unity), devemos realizar as seguintes ações:

1. Crie um objeto de jogo vazio.

2. No Inspetor, clique em Adicionar componente → Gerenciador de rede.

3. Depois disso, selecione uma Camada de Transporte no componente Network Manager.

4. Uma camada de transporte estabelece comunicação entre seu aplicativo e diferentes hosts em uma rede.

5. Vá em frente e selecione o Unity Transport mais recente. Isso adicionará automaticamente o componente Unity Transport ao Network Manager.

Segue abaixo a reprodução de como ficará na cena:

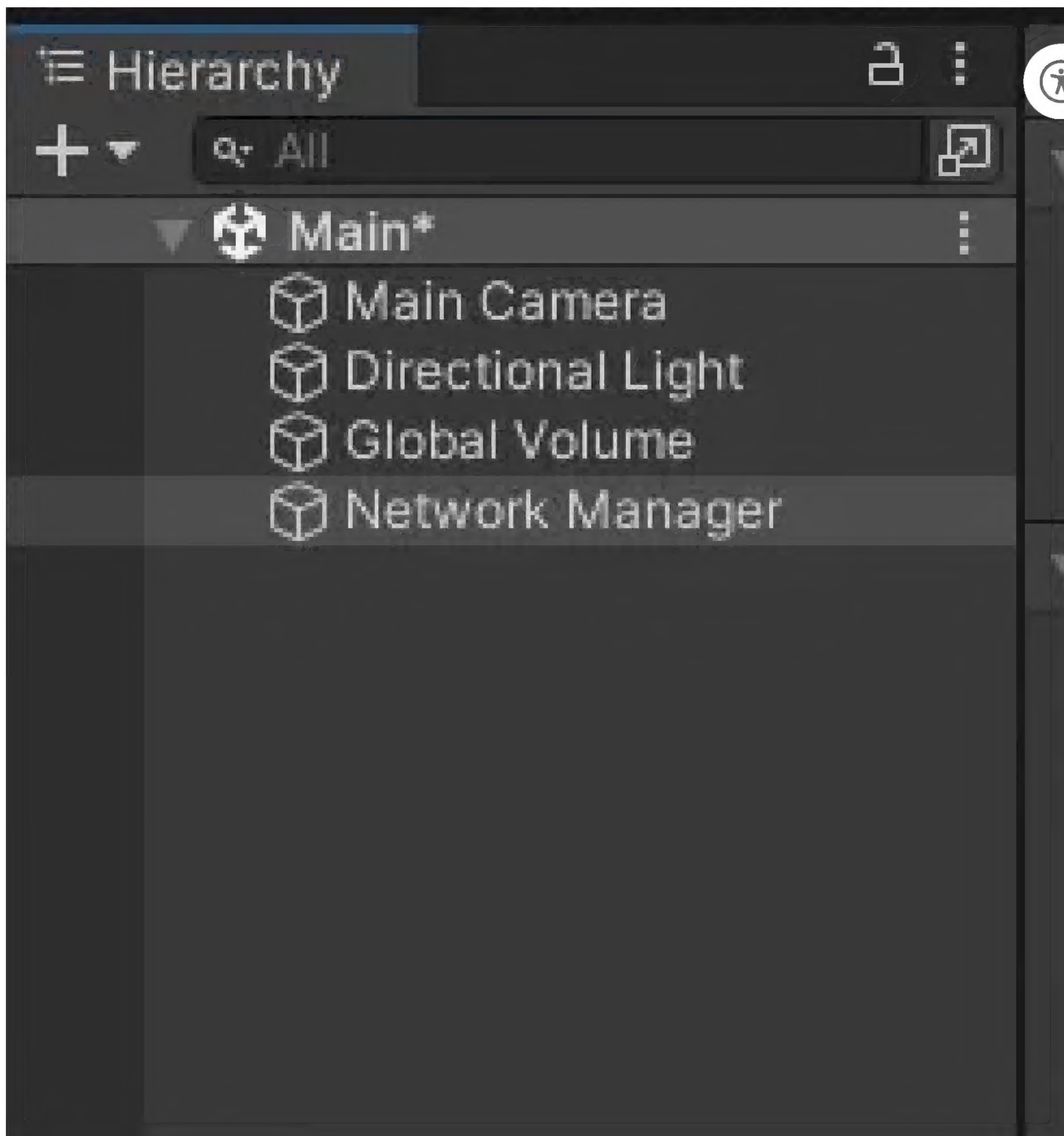


Figura 1 – Reprodução da Unity

Fonte: O próprio autor, 2023.

A partir desse contexto, temos o modo multijogador instalado e funcionando. Devemos finalmente criar o Player **prefab** que será gerado quando qualquer cliente entrar.

4º. Passo) Desenvolva o roteiro a partir da manipulação da Unity e na leitura da documentação oficial do Netcode, a fim de criar o Player prefab do jogador que será gerado quando qualquer cliente entrar. Descreva as ações necessárias que devem ser adotadas na Unity e construa o roteiro de

desenvolvimento com as informações necessárias que o programador deve seguir, bem como alguns prints (imagens) para demonstrar o que está sendo realizado durante os processos.

Gabarito Atividade Prática: O Netcode pode gerar um objeto de jogador padrão para você. Se um Player Prefab válido for atribuído na Rede, o Netcode gerará uma instância única do player prefab designado para cada Cliente conectado e aprovado. Seria possível, por exemplo, representar o Player a partir da inclusão de uma cápsula. A cena deveria se parecer com a imagem abaixo:

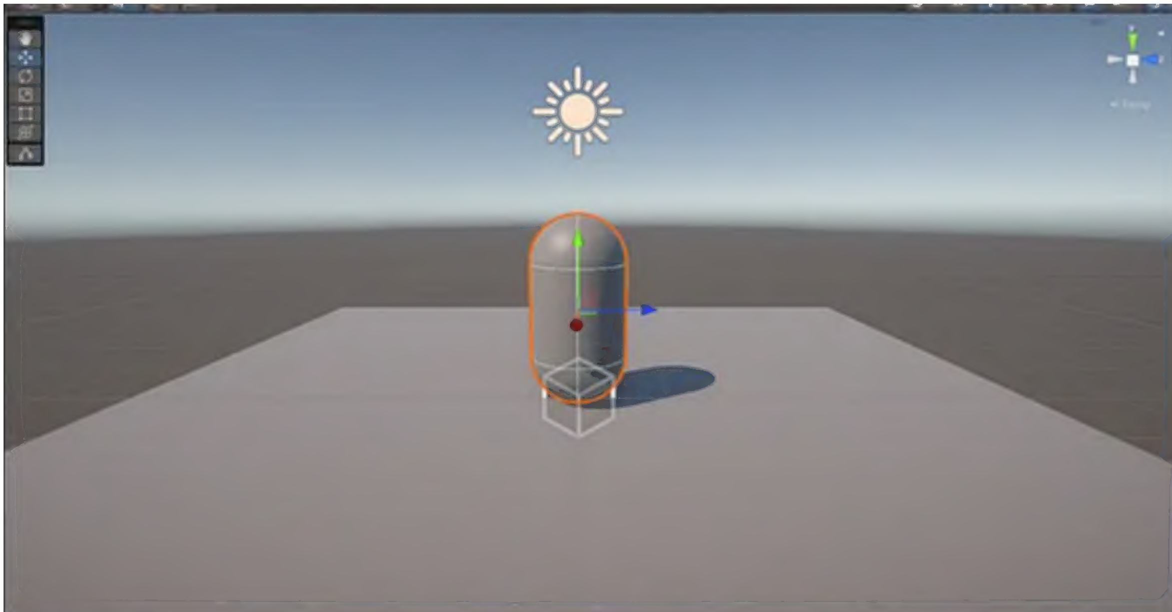


Figura 2 – Reprodução da Unity

Fonte: O próprio autor, 2023.

Após esse processo, seria necessário criar uma pasta "Prefabs" e arrastar e soltar o objeto Player da cena para essa pasta para torná-lo um prefab. Em seguida, deveríamos excluir o objeto da cena.

Após essa etapa, será necessário vincular esse prefab com o componente Network Manager. O processo é muito simples, exigindo que o programador arraste e solte o prefab que foi criado. O Gerenciador de rede

gerará o pré-fabricado do Player atribuído sempre que um novo cliente se conectar.



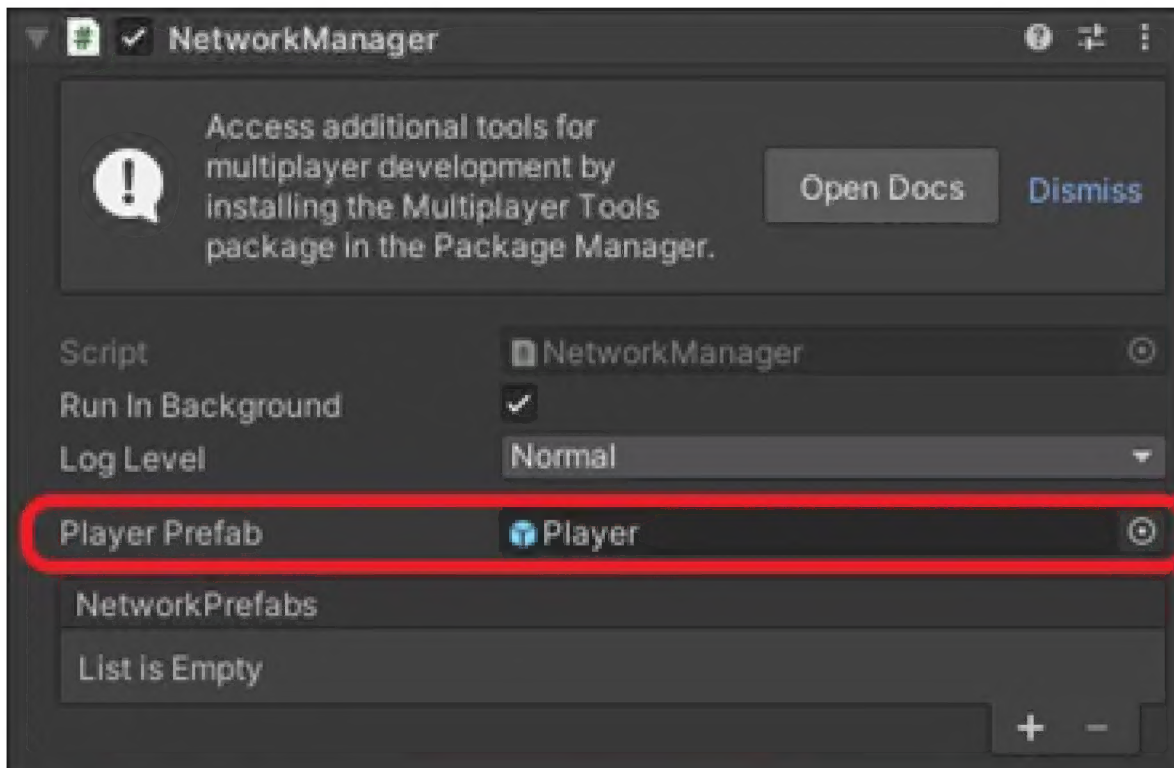


Figura 3 – Reprodução da Unity

Fonte: O próprio autor, 2023.

Por fim, devemos adicionar o componente NetworkObject ao prefab do Player.

Ir para exercício